ACTIVE NETWORKS: SAFETY ISSUE

Devi Krishnan

Dr.Sureswaran Ramadass
Distributed Computing
School of Computer Sciences
Universiti Sains Malaysia
11800 Penang, Malaysia
{devi@nrg.cs.usm.my}

Abstract: Safety and security ar two most important properties of a system. A safe system provides protection against error of trusted users, meanwhile a secure system protects against error introduced by untrusted users. In this situation, high requirement for rapid service creation have stimulated the development of programmable network infrastructure where end users or service providers can customize the properties of a network infrastructure while it continues to operate. The ability on customization of network infrastructure properties is called active networks[2]. But the main concern of potential users of such systems is their reliability and most specifically their safety and security. As for this paper, the scope of the research is towards the safety issues involved in the active network architecture. A variety of proposals for exposing some control of network infrastructure have been made. Priority has been given on exposing this shared infrastructure to users must preserve some expectations of reliability while allowing the infrastructure to be multiplexed to derive the economics advantages of sharing. In this paper, attention has been given to one of this proposal, discuss further on the plus and minus point of it and also the future suggestion towards the better performance of the active networks.

Keywords: networking, active network, security aspect of active network, safety issues

1 Introduction

1.1 Motivation

Active Networking is an exciting area of research which concentrates on two commonly separated approaches programmable devices[2], and capsules[3]. These approaches can be viewed as the two extremes in terms of program ode injection into network nodes. Programmable switches typically learn by implicit, out-of-band injection of code by a network administrator. Research in the area of programmable switches either focuses on how to upgrade network devices at run time or on upgrades which support end systems applications for example congestion control for real time data streams or on a combination of both.

Capsules, though are miniature programs that are transmitted in-band and executed at each node along the capsule's path. This approach introduces a totally new paradigm to packet switched networks, instead of passively forwarding data packets, routers execute the packet's code and the result of that computation determines what happen next to the packet. It looks like this approach has an enormous potential impact for the future of networking. In the near future, capsule-based solutions potentially suffer from performance related problems mainly due to security constraints. They commonly make use of a virtual machine that interprets the capsule's code to safely execute on a node. This is similar to the way Java applets run in web browsers. The virtual machines must restrict the address space a particular capsules might access to ensure security, which restricts the application of capsules.

One of the motivation of active networks is to reduce the difficulty of integrating new technologies and standards into a shared network infrastructure. If one follows the development and deployment efforts in the context of the Ipv6[4] network protocol realizes how extremely difficult such a project is in today's heavily used internet.

1.2 Problem Statement

In this environment discussed above, since it is a specific independent of architecture, the designer of a network has a set of tradeoffs they must make which define a design space. There are five main issues to be discussed here. First flexibility because it is a measure of the system to perform a variety of tasks. Usability since is a measure of the ease with which the system can used for its intended tasks. Third performance, how the system will have some quantitative measures by which it is evaluated such as throughput, delay and delay variation.

Evaluation on cost because networking system will have quantifiable economic costs, such as costs for construction, operation, maintenance and continuing improvements. Finally, safety the most important and going to be the discussion in this paper. Since network systems are shared resources the designer must provide mechanisms to protect users from each other to a policy. If the safety issues is designed in, it can simply be made part of the design specie in which are attractive cost and performance tradeoffs.

Safety and security are two reliability properties of a system. A safe system provides protection against errors of trusted users, while a secure system protects against errors introduced by untrusted users. So there is considerable overlap between mechanism to support each property. The main concern of the discussion in this paper will be the safety issues involve in the development of active networks and more ideas in the improvement of the safety which is to protect active network against errors of trusted users.

There have been few protocols has been introduced to provide the best safety to the programmable network infrastructure. This paper will mainly study on the SANE environment and discuss more on how the environment could protect the devices and how it can be improved.

2 Literature Study

2.1 Introduction to Active Network

Active networks allow individual user, or groups of users, to inject customized programs into the nodes of the network. "Active" architectures enable a massive increase in the complexity and customization of the computation that is performed within the network, for example, that is interposed between the communicating end points.

Active networking is based on programmable intelligence in network nodes (switches and routers) consisting of processing and memory beyond that conventionally associated with packet forwarding[4]. Programs can be dynamically injected into the network for execution on active nodes. The concept of active networks represents a new approach to computer networking. In an active network, packets are not just forwarded by a router as in the current Internet instead, a router in an active network has the capability of processing the payload contained in a packet in an arbitrary, application-specific way. Therefore, active networks will allow service providers or even users to tailor the functionality of a network to their needs, by placing application-specific code into the network. Thus, active networking is a technology that can be used to implement programmable networks - networks providing a basic hardware and software infrastructure, which can be dynamically adapted to the needs of the service provider or the user. They will allow dynamic service creation and deployment, just in time approach to providing services in a future Internet.

Active Network exploits programmable infrastructure to provide rapid and specialized service introduction. The main purpose of active network is to make the network elements programmable, provide flexibility network functionality, avoid the lengthy standardization processes and also compatible with the legacy networks. If the traditional protocols rely on prior agreements of message processing between communicating parties but active network is different in a way that it relies on prior agreement of a computational model. Basically active networks conceptualize networks as a

collection of a “active node” and a collection of “active packet”. Active node here is programmable network element that can perform any computational on it and active packet is network message that carries programs.

Active networks are packet-switched networks in which packets can contain code fragments that are executed on the intermediary nodes[1]. The code carried by the packet may extend and modify the network infrastructure. The goal of active network research is to develop mechanisms to increase the flexibility and customizability of the network and to accelerate the pace at which network software is deployed.

Increase flexibility and customizability implies security problems. Stability is crucial to network devices not only because the network has become so important to people’s daily work but also because the devices lie in separate administrative domains and are run by different people. Breakdowns are extremely hard to trace and fix.

Among all these reason for active network, the most important application of active networks stems directly from their ability to program the network new protocols and innovative cost-effective technologies can be easily employed at intermediate nodes. The functions of the nodes will no longer be rigidly built-in by vendors who must follow designs dictated by slow and intractable standards committees. Also, network integrity will not be vulnerable against various ad hoc approaches toward network programming, as is the case today.

At the same time, active networks can be very beneficial for a variety of specific applications for example network management, congestion control, multicasting, and caching.

2.2 Safety Issues Involved

So far, we have explored various fields of networking where active networks can be useful. There are few security and safety issues that active networks raise. Since active networks are much more flexible than passive, the number of safety and security issues that need to be addressed are tremendously increased. By safety we mean reducing the risk of mistakes or unintended behavior. By security we mean the usual concept of protecting privacy, integrity, and availability in the face of malicious attack. A packet that carries executable code can potentially change the state of a node. Nodes (routers, switches, etc.) are public resources and are essential to the proper and correct running of many important systems. Therefore, the safety and security requirements placed upon the computational environment where the code of packets will be executed must be very strict. In the current Internet, the only resource consumed by a packet at a node is the memory needed to temporarily store it and the CPU cycles necessary to find the correct route. In such an environment, strict resource control in the intermediate nodes was considered non-critical. However, an active packet may consume not only many more resources but also at a faster rate. Denial of service attacks may easily occur if there is no resource management. Clearly, in addition to security and safety, fairness is also an issue.

In an active network, active packets may misuse active nodes, network resources, and other active packets in various ways. Also, active nodes may misuse active packets. Previous work related to the security issues of mobile software agents is directly applicable here [9]. Some of the possible problems that may occur are the following:

- Damage: An active packet can destroy or change the resources or services of a node by reconfiguring, modifying, or erasing them from memory. A node may erase an active packet before the completion of its job in the node. Finally, active packets that share the same computational environment may attack each other.

- Denial of Service: An active packet may overload a resource or service due to constantly consuming network connections or using a great portion of the CPU cycles available. The

node cannot function properly under these circumstances and another active packet cannot be executed or forwarded.

- Theft: An active packet may access and steal private information from a node. On the other hand, an active packet is vulnerable toward the node at any point when visiting it. Even if it is encrypted, it is not totally safe because it usually has to be decrypted in order to execute.

- Compound attack: The biggest actual threat for an active node is a compound attack aimed toward a goal. For example, a malicious user may send many active packets toward a central router and try to bring it down by consuming all its bandwidth capacity.

Protecting the nodes and the packets in a flexible environment such as active networks is not an easy task. Some techniques that may be used to protect the active nodes are :-

- Authentication of Active Packets: Any active packet should have authenticating credentials produced using one of a number of algorithms such as a public key signature algorithm. This do not guarantee that the active packet will be harmless, or even useful. Credentials only provide assurance that someone else vouches for the active packet.
- Monitoring and Control by using a reference monitor. A reference monitor may be used to restrict the information, system resources and services that active packets are allowed to access and use. The reference monitor consults a security policy to determine if access is to be granted. Since access-level monitoring places restrictions directly on what a packet can do, it is an effective method. However, the decision of granting permission for using some resources is based upon some credentials which are not able to guarantee that a packet is harmless as it is already mentioned.

- Limitation Techniques for example time limitation. Time limits such as the amount of time an active packet may be allowed to be executed, range limits such as the total number of nodes the packet is allowed to traverse, as well as duplication limits for an instance the number of times that a packet may duplicate itself, are essential in preventing an active packet from monopolizing the resources of a node.

- Proof Carrying Code (PCC)[5] is based on the observation that is often easier to check an answer than to produce it. For a mobile program, it is the creator of the program who knows the key reasons it is correct, not the host active node that receives the program. Hence we could pair the mobile program within each active packet with a proof of its correctness. The active node may easily check the proof and then run the program. The difficult part is the creation of the proof but this is the job of the program creator.

Two methods are suggested for the protection of the active packets are fault tolerance techniques and encryption. Encryption refers to the situation where active packets do not consist of clear text code and data[6]. Encryption is usually used for code and data in transit. However, the programs may even be executed in a non-cleartext form, which leads to the concept of mobile cryptography [7].

The fault tolerance techniques are replication, persistence, and redirection. Replication means that packets replicate at each node. Persistence means that packets are temporarily stored against node failure so that even if a node crashes, the copy persists in storage. Redirection means that packets may seek alternative routes in case their default route fails.

Replication and persistence are unacceptable for the vast majority of network packets because they consume memory and bandwidth, and only very important active packets should be allowed to do this for example packets installing a new version of a routing protocol in all nodes[4]. Redirection and encryption have broader applications in packet protection because they basically consume CPU cycles. A combination of fault tolerance techniques and encryption may give very good results in the problem of protecting active packets. However, because these techniques are still in their infancy,

there is much to be done before definite results are reached.

3 Discussion

Active networks are aimed at incorporating programmability into the network to achieve extensibility. An approach to obtaining extensibility is based on downloading router programs into network nodes. Although promising, this approach raises several critical issues: expressiveness to enable programmability at all levels of networking, safety and security to protect shared resources, and efficiency to maximize usage of bandwidth.

From the description of the active network, we can see that there are many entities in an active network that have assets that we would want to protect. The end user at the source and destination, the active node itself, the execution environments and the active code/domain all have security concerns[5]. The end user retains the traditional concerns about the authenticity, integrity, and condentiality of the packet's payload data as it traverses the network. As the active code may create persistent state in the active nodes it traverses, the end user will have the same concerns about data created in the infrastructure as well as concerns over access to that data.

The active node's security concerns are likely to be concentrated on authorization of use of the node's services and resources, in order to maintain availability of use. It will, of course, also be concerned about the integrity and condentiality of its own state. The active code where standing as proxy to the end user who launched the packet and has safety concerns that are related to access to its services for example access to the domain in which it is executing and access to sharable persistent state it creates.

End user viewpoint where the end user would rather not have to trust all active nodes, execution environments, and other active code in the active network. Results from certain research areas regarding mobile agents which is running on untrusted hosts, there are few ways to assure the end user that its data will be protected from attacks, exposure, unauthorized use by the node in which its packets are processed in the clear. The end user may apply end-end cryptographic protections against these attacks and not make the node so that the data is not in the clear in the node.

While end-end cryptographic protections limit the damage that the node and can cause to the data, they do limit the network services that can be performed for the packet. The only other assurance the end user has that it can be protected against attacks by the node from an ability to direct the active code to avoid transmitting the packet to untrusted nodes or execution environments.

The end user has some method of identifying nodes are trusted and authenticating the nodes with the active packet encounters, and that the packet transmission will be under the exclusive control of the active code. The safety node can provide enforcement of the end user's authorization policy, as long as they have the ability to authenticate the principals associated with each active code and are provided the end user's policy.

Node viewpoint where the node has its own view of the threat sources in the active network. It should not be necessary for the node to completely trust the node it executes. It would certainly be unwise to architect the system so that it must trust the active code it runs or the end users who generate packets. Therefore, the node would view the active code, and the arriving packets as potential threat sources.

Because the node architecture grants right to start subdomains if and when it wants, requests from the executive environment for node[4] services might be intended to provide services for a packet not assigned by the EE to a subdomain. Such a request will be judged by the node on the basis of the privileges granted to the executive environment's domain. The NodeOS must trust the executive environment to properly use its own privileges on behalf of active code that is not assigned

to a subdomain. That is, the node must trust that the executive environment is adhering to the node authorization policy in requesting node services on behalf of an active code.

Because the node has control over the allocation of resources and privileges to an executive environment's domain, it has the opportunity to mitigate the possible damage from an executive environment. It can balance the trust it holds in an executive environment with a judicious allocation of resources and privileges. Fully trusted executive environment's might be provided with more resources and more powerful privileges than less trusted executive environment's. The threat from active code can be controlled because the node has the opportunity to enforce its own authorization policy for the actions of any domain. Finally, countering clogging attacks from arriving packets is a research area of its own. In short, protection against clogging attacks requires that the node's neighbors cooperate with limits and that the node establish limits with its neighbors that in the aggregate do not exceed its capacity.

Execution Environment Viewpoint where the environment sees the same threats from active code and arriving packets as the node sees. It has the same opportunities to control the threat from active code by enforcing its own policy governing access by active code. The execution environment can rely on the node to enforce the execution environment's policy governing acceptance of arriving packets, as long as its required authentication of the packets is within the capabilities of the node, for example does not require some execution environment specification authentication mechanisms, and the node is provided with the policy. The environment sees potential attacks from other environment's through shared persistent state or access to its services.

As these access methods must be provided by the node, the environment must rely on the node to enforce the policy or attributes that are important. In active networks, the aspects of the principal's identity that are important may change radically as the packet traverses the network. Within the end user's enterprise network, the individual's identity or company role may be important. But beyond the immediate network of the end users, it is not likely that the individual identity of the end user will be important. Aggregate security attributes will be more likely to be used, which may be labels, groups and others[4]. Furthermore, the aggregate attributes may themselves differ in different domains.

As for the active code viewpoint the active code itself would rather not have to trust all the nodes, execution environment's and other active code in the network. Unfortunately, it is in the same situation regarding trust in the node and execution environments in which it runs as the end user is. The active code must trust the nodes which it executes and avoid those itdoes not trust. The active code sees potential attacks from other active code through access to shared persistent state or services. These access methods are provided by the environment and so the active code can rely on.

3.1 Protection Techniques

Protection techniques that active network community employs two ways of ensuring that possible attacks are avoided. The first is to limit the possible actions to those that would be safe for any entity in the system to perform. These are language based approaches, involving type-safe and namespace limiting languages. This is a low cost technique with a large payoff

But it is not always possible to eliminate dangerous activities entirely. There will be some actions that some, but not all, entities should be permitted to perform. The second class of techniques associates a principal with each request for an action and enforces a policy that states which principals are permitted to perform which actions. These are authorization based approaches. In this work we have concentrated on authorization enforcement to protect active networks and the authentication to support authorization enforcement.

3.1.2 Authentication Challenges

Some of the most challenging aspects of securing active networks concern the authentication support for authorization[5]. Authorization decisions require the authentication of the entity making a request. Authentication normally implies the use of cryptographic techniques[8]. But the application of existing cryptographic techniques to the active networks environment presents certain challenges. First, the identification of the principal itself in active networks is challenging. Existing Internet interactions are typically client/server, where the explicit individual identity or attributes that are important.

Second, the choice of an authentication mechanism presents challenges in active networks. Existing mechanisms for providing authentication protection of a packet are rooted in the existing Internet paradigm of client and server based communication. These will not be sufficient in an active network environment where the packet needs to be authenticated at source and destination and potentially every node in between. The existing solutions can be used hop-hop in the path but that provides little in the way of end source authentication. If all nodes in the active networks can be trusted and the edge node correctly identi.es the end source, then hop-hop protection provides sufficient authentication. However, experience in the Internet for example wide-spread Internet outages caused by one faulty router which has provided ample proof that it would be folly to trust all nodes as a set.

Existing solutions can also be used to set up multiple security associations, one between the end user and each node on the path. The latency and bandwidth requirements to establish each association and multiply protect the packet would be prohibitive. Even though hop-hop protections do not provide strong end-end authentication, hop-hop integrity protections are still important. Integrity protection between neighboring active nodes provides protection against attacks from outsiders, and should include protection against replay, modification and spooling. This first level of protection is particularly important in neighbor to neighbor exchanges or signalling[8].

When hop-hop protections do not provide sufficient end-end authentication of the principal associated with a packet, we can employ end-end protections. However, the use of end-end cryptographic techniques is also a challenge in active networks. Symmetric techniques could be used if a key associated with the principal could be installed at each node of the packet's path through the network. The packet modifications at each node could be protected anew with the shared key. However, this has a similar trust drawback as using hop-hop protection for every node on the path must be implicitly trusted.

Also, the assurance of authenticity of the principal, derived from the shared key, is diluted if the key is not unique to the principal and the path. For the strongest assurance, each new communication would require key distribution or agreement among the nodes of the path. This expensive operation would be unsuitable for a datagram model of communication and would motivate not only a connection oriented model for communication but a virtual circuit model, where all packets in a row of packets are protected by the same key and transit the same nodes.

Asymmetric techniques for example digital signatures can operate in a datagram model but have difficulty protecting packets that change. Signing a packet with a digital signature provides a cryptographic association from the signer to every potential verification of the future. Therefore, authentication by digital signature is suited for a datagram model of communication, where the packet may decide in route what nodes it will visit. The digital signature protection provides the strong end source authentication that we wanted. However, the asymmetric private key that is used to produce the signature should be kept secret by the signer to maintain the security features of a digital signature. This means that the private key would not be known to the infrastructure nodes that process the packets. Consequently, infrastructure nodes cannot produce a new end source signature if the packet is modified in transit. It might be possible to have each modifying node sign just the modifications it

makes, but such a scheme produces massive packet growth. Finally, the performance issues with asymmetric cryptographic techniques in terms of both performance and bandwidth are well known.

4 Conclusion

This paper has made two constribution towards the topic discussed. First it clearly explain the architecture or how an active networks work. Why active network is important and second issues discussed throughout the paper is on how to implement the safety features in active network to make sure the flexibility of this system is not misused. The study has given the clearer picture of an active networks and its challenges.

Acknowledgements:

I would like to express my heartfelt thanks here for the support I received from many people from Universiti Sains Malaysia.

In particular, I would like to express my deep appreciation to Dr. Rahmat and Dr. Fadzilah Harun for their insightful guidance and dedication to supervise this project. Then again, thanks to Dr.Sureswaran Ramadass for careful examination of this paper.

References

[1] David L. Tennenhouse and David J. Wetherall. Towards an Active Network Architecture. Multimedia Computing and Networking. (1996)

[2] D. L. Tennenhouse *et al.* A Survey of Active Network Research. *IEEE Commun. Mag.*, vol. 35, no. 1. (1997).

[3] D. Scott *et al*. A Secure Active Network Environment Architecture. *IEEE Commun*. (1998).

[4] S. Bhattacharjee, K. L.Calvert, and E. W.Zequra. Implementation of an Active Networking Architecture. (1996).

[5] S. Bhattacharjee, K. L.Calvert, and E W.Zequra. An Architecture for Active Networking. *IEEE INFOCOM '97*. (1997).

[6] Beverly Schwartz, Wenyi Zhou, and Alden W. Jackson. Smart Packets for Active Networks. BBN Technologies. (1998).

[7] Albert Banchs *et al*. Multicasting Multimedia Streams with Active Networks. ICSI technical report. (1997).

[8] Michael S. Greenberg, Jennifer C. Byington, and David G. Harper. Mobile Agents and Security. *IEEE Commun. Mag.* (1998).